English translation and summary of the contents of Fabian Streitel’s Bachelor’s thesis, to help improve SAMM after comparing it to BSIMM and other models

**Definitions**

* Secure Software Lifecycle: Attempt to cut cuts by fixing root causes to problems in software as early as possible
* Process Model: description of a list of quality requirements that are desirable in a process
* Maturity Model: Process Model with a definition of maturity levels, that build upon each other and which determine its adequacy for a given purpose.
* Policies: Abstract objectives for topics that are of relevance throughout the organization
* Standards: Define a recognized way of complying with a policy
* Guidelines: Technical rules/prescriptions that issue specific instructions that must be observed.
* Best Practices: Technical recommendations, based on experience, that have proven beneficial for a given purpose/objective.

**Possible Areas to Improve:**

* Awareness program and mentoring/monitoring
* Network Security
* Penetration Tests
* Attacker's perspective
* Top N vulnerability list, closer to BSIMM
* SW Development Guidelines, like in MSSDL
* Not very concret and measurable. Too short in the description
* Success metrics seem to be very arbitrary. Difficult to define values for new metrics to build in
* Personnel: On which basis were the values set? They seem very arbitrary. It's hard to tell how much effort a new activity introduces in the organization.
* Organizational Structure: not applicable for all organizations. What about agile methods or very big companies with more hierarchical layers, or very small companies that don't have that many people?
* Better definitions like Policy, Standard, Guideline, Best Practice, design review, Risk and Threat analysis
* Examplary case
* Improve topics: "Results", "Personnel", and "Success measures" to make them less arbitrary

**Suggestions:**

* **PENTESTS**
  + Improve differentiation between penetration and security tests.
  + As it is used, it means more security test than pentest
  + From BSIMM:
    1. use external pentesters (manual pentests)
    2. Automatize penetration tests (use pentesting tools internally)
  + Title: Conduct penetration tests before software releases
  + Description:

Use internal or external experts with experience to conduct penetration tests on the application, shortly before its release. This should preferably be done in a test environment that mimics the production situation as closely as possible. If there is no such environment, test the system in production, but ensure its safety and integrity.

The goal of the penetration test is not to validate that a fixed set of security tests are met, but rather to uncover previously unknown or overlooked vulnerabilities in the application. This warrants an explorative approach to testing where the tester searches the system for any kind of possible abuse of functionality. The more the tester knows about the application, the more likely they are to find problems. Thus, provide them with all available documentation on the system (e.g. design documents, security requirements, etc.)

Perform this test before each new release of the software. If possible, have stakeholders review the penetration test results before putting the application into production.

* Part of Security Testing
* Results: The explorative approach allows for the detection of additional vulnerabilities

that have been overlooked.

Penetration tests that are carried out by external experts or specialized security auditors provide an independent verification of the application’s security posture.

* Costs: Once-per-release project overhead from penetration testing
* No related levels
* **Network Security**
  + Title: Perform network-level analysis
  + Description: From CLASP 4.3.2 and 4.5.1

Considering the application’s architecture and design, make a list of all network interfaces and components that the software relies upon, e.g. required ports and connections, firewalls, web servers, databases etc. Furthermore, note down network components that must not be present when the application is deployed. Finally, note down all other network systems that will have an impact – be it positive or negative

– on the operation of the software.

Take the gathered information and compose a diagram that depicts the network infrastructure the application expects when being deployed. This should contain all of the necessary components identified above. Finally, mark all trust boundaries, i.e. components which separate different areas of trust, e.g. firewalls or servers.

All of this information must be documented in the operational environment

specification and should be updated on any changes to the

application’s design.

* Part of Environmment Hardening, level 2
* Results:

The composition of the network design diagram provides insight into the network architecture and can reveal problems, false assumptions and conflicting requirements.

The network diagram allows for a quick understanding of the network architecture and design decisions.

* Costs: Ongoing project overhead from maintenance of network design documents
* Related levels: Environment Hardening 1 & 3
* **Creation of a Security Policy**
  + Title: Create global security Policy
  + Description: from PC2A and CLASP (remove it from PC2A)

Gather a list of non-compliance security requirements that should apply to every project and form a baseline security posture. Differentiate according to existing project classifications where necessary, e.g. prioritize requirements according to the risk level of the project or omit requirements based on the project or deployment type. Also consider existing policy within the organization when gathering and

prioritizing requirements.

Such a document can be assembled from known business drivers for security or online resources on security policies. The simplest method is often to consult existing guidance being provided to development staff and gather a set of best practices. Avoid making choices that are arbitrary and potentially limiting.

When the requirements for a new project are specified, consult the security policy and adopt all mandatory items from the list, adding them to the specification. Use the list to document any omission of and deviation from the policy and the reasons behind that choice. This document can be used in audits to check for the coverage of

the security basics.

The policy should be reviewed regularly. Projects that have been initiated since the last review should be inspected for new requirements that can be incorporated into the policy.

* Part of Policy and Complience, belongs next to PC2A at level 2
* Results:
  + Project teams know the baseline security requirements.
  + The policy eliminates common security problems, regardless of the specific platform and programming language in use.
  + The policy establishes good security practices within the organization.
* Costs: Initial creation and ongoing maintenance of security policy
* Related Levels: PC2A
* **Taking the attakers perspective**
  + Title: Create an internal forum to discuss attacks
  + Description: from BSIMM AM2.4 with some modifications

Establish an organization-wide means of communication, e.g. a mailing list or an internal forum. If possible, utilize already extant resources and platforms within the organization.

Once created, advertise it as a means of discussion of the attacker’s perspective throughout the organization. This can be done, for example, by sending a mass mail or writing an article for the company newsletter.

The content of the forum should be focused specifically on attackers and their methods and motivation. Security-savvy staff should be selected and assigned the task of getting discussion started by composing interesting and engaging posts and publishing them on the forum. Furthermore, links to online references/tutorials or books, which teach the attacker’s perspective, can be accumulated in order to give interested people a quick introduction to the topic.

* Part of Education & Guidance at level 2
* Results:
  + Interested staff can discuss the attacker’s perspective internally.

The discussion improves the security expertise within the organization and raises awareness for the importance of out-of-the-box thinking.

* Costs: Ongoing maintenance and change control costs for the forum
* Related Levels: -
* **Creation of Development Guidelines:**
  + Take from MSSDL : Use the Latest Compiler and Supporting Tool Versions und Use Defenses Added by the Compiler.
  + Title: Use up-todate language-specific security tools
  + Description:

Compile a list of all the development tools used in a project. The list should include all compilers, interpreters, code analysis tools etc. Next, for each tool note all security related functionality it provides, e.g. compiler flags that enable warnings or add additional security checks, sandboxing features etc.

From that list, choose features that make sense for the project and are easy to use. Focus should be on immediate security gain with minimal disruption of the development process. Over time, more features can be added to the process. For each chosen feature, create a security requirement that demands the use of that feature during development, as well as specific technical guidance on how to do that using the respective tool. Such a requirement might, for example, demand that code that is checked into a central repository must compile without warnings of a certain level.

Advertise those guidelines among development staff and publish the new security requirements. Update the list of required features regularly. In order to ensure that newly developed security features can be used, require an update of all tools to newer versions regularly.

* Part of Security Requirements at level 1
* Costs: Costs for development tool updates
* Results:
  + Using up-to-date tools allows the usage of state-of-the-art security features.
  + Enabling security features by default increases the chances of finding vulnerabilities and bugs during development.
  + Creating guidelines for developers on how to use the security features for their platform allows for easy and correct integration of those features into the development process.
* Related levels: Education & Guidance 1
* Address PII
* Vendor Contracts
* Board shall be aware of regulatory conformance